SMS Spam Detection System Using NLP

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Yagya Raj Bhatt, yagyabhatt57@gmail.com

Under the Guidance of

Abdul Aziz Md

Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to everyone who supported me during the course of this project. Their guidance and encouragement were instrumental in the successful completion of this work.

Firstly, I extend my heartfelt thanks to **Abdul Aziz Md**, Master Trainer, Edunet Foundation, for his invaluable mentorship and guidance throughout this project. His advice, critical insights, and continuous support were crucial in shaping the direction of this work and in overcoming challenges along the way. His expertise and encouragement have been a constant source of inspiration.

I also express my gratitude to **AICTE Internship on AI: Transformative Learning** and **TechSaksham – A joint CSR initiative of Microsoft & SAP**, for providing me with this incredible learning platform. The program offered valuable resources, training, and opportunities to enhance my understanding of AI and its transformative potential.

Lastly, I thank my peers, family, and friends who supported me directly or indirectly during this endeavour. Their patience, encouragement, and unwavering belief in my abilities were essential in completing this project successfully.

ABSTRACT

The increasing prevalence of spam emails poses significant challenges to email users, as these messages often disrupt communication and expose individuals to potential security risks. To address this issue, this project aims to develop an efficient **Spam Email Detection System** leveraging machine learning and natural language processing (NLP) techniques. By classifying emails as either "Spam" or "Not Spam," the system provides a robust solution for automated spam filtering.

The project utilizes the **SMS Spam Collection Dataset**, which contains labelled email and SMS content, to train a classification model. The dataset was meticulously pre-processed through steps such as text cleaning, removal of special characters, and tokenization. Additionally, class imbalance in the dataset was addressed using oversampling techniques to ensure fair training of the model. For feature extraction, the **TF-IDF Vectorizer** was employed, transforming textual data into numerical representations suitable for machine learning algorithms.

A **Logistic Regression** model was selected for its simplicity, efficiency, and reliability in binary classification tasks. The model achieved an accuracy of **92.06%**, with a balanced performance in identifying both spam and non-spam emails. To enhance user accessibility, a single-page web application was developed using **Stream lit**, offering an intuitive interface for inputting email text and receiving real-time predictions. The user interface incorporates visually distinct elements, such as color-coded result displays, ensuring clarity and usability.

This project demonstrates the successful integration of machine learning and NLP to tackle real-world problems in email communication. The system is scalable and can be expanded to detect spam in multiple languages or integrated with popular email services for live filtering. The results highlight the effectiveness of the system, making it a valuable tool for users seeking to streamline their email experience and mitigate the risks associated with spam emails.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Problem Statement:

Spam emails, characterized by unsolicited and often harmful messages, have become a growing challenge in the digital age. These emails not only disrupt the flow of communication but also pose significant security risks, including phishing attacks, malware distribution, and financial fraud. Traditional rule-based systems designed to detect spam often fall short in addressing the dynamic and evolving nature of spam content. As spammers continually adapt their strategies to bypass filters, these systems generate high false positives and negatives, either blocking legitimate emails or letting spam through. This inefficiency wastes users' time and resources, creating a pressing need for a more adaptable and intelligent solution.

The significance of addressing this problem lies in its far-reaching consequences for individuals and organizations alike. Spam emails impact productivity, reduce trust in email communication, and expose users to substantial financial and data security risks. With the increasing reliance on email as a primary communication tool, the demand for a reliable and scalable spam detection system has never been more critical. By leveraging machine learning and natural language processing, this project aims to overcome the limitations of traditional systems, providing a robust solution that not only improves detection accuracy but also reduces the burden on users, ensuring a secure and seamless email experience.

1.2 Motivation:

Spam emails have become a pervasive issue in digital communication, consuming significant time and resources while exposing users to security threats like phishing, malware, and fraud. Traditional spam detection systems often fail to adapt to evolving spam tactics, resulting in inefficiencies and inaccuracies. The motivation behind this project lies in addressing these challenges by developing a robust and scalable solution that leverages machine learning and natural language processing (NLP) to deliver accurate and automated spam classification. This ensures enhanced productivity, improved security, and a seamless user experience.

The objective of this project is to design and implement an intelligent **Spam Email Detection System** capable of accurately classifying emails as "Spam" or "Not Spam" using machine learning and natural language processing techniques. The project focuses on preprocessing email content to extract meaningful features, training a reliable classification model to improve detection accuracy, and developing an interactive, user-friendly web interface for real-time predictions. By addressing the limitations of traditional spam

detection systems, this project aims to enhance email security, reduce false positives, mitigate user exposure to malicious content, and streamline email communication for both individuals and organizations.

Applications:

- **Email Service Integration**: The system can be integrated into email platforms like Gmail or Outlook for real-time spam filtering, reducing the user's burden of sorting emails manually.
- **Business Communication**: Organizations can use this system to ensure the smooth flow of critical communications by minimizing interruptions caused by spam.
- **Cybersecurity**: The system can act as a frontline defense against phishing attempts, protecting sensitive information and preventing financial fraud.
- **Scalable Deployment**: The project can be extended to detect spam in messaging platforms or social media, broadening its utility beyond email.

Impact:

- **Improved Productivity**: By automating spam detection, users save time and effort otherwise spent sorting through emails.
- **Enhanced Security**: The system helps mitigate risks associated with spam, such as phishing attacks and malware infections.
- **Scalability**: As the system is scalable and adaptable, it can address the dynamic nature of spam and expand to diverse use cases, including multilingual detection.
- **Accessibility**: With its intuitive user interface, the system ensures ease of use, making it accessible to a broad range of users.

1.3 Objective:

The objective of this project is to develop a robust Spam Email Detection System that can accurately classify emails as either "Spam" or "Not Spam" using machine learning and natural language processing (NLP) techniques. The system aims to address the challenges posed by traditional rule-based spam filters by leveraging advanced methods for preprocessing email content, extracting meaningful features through TF-IDF vectorization, and training a Logistic Regression model for reliable predictions.

Additionally, the project focuses on creating an interactive and user-friendly web application using Streamlit to provide real-time email classification. This system aspires to improve email communication efficiency, enhance security against phishing and malicious emails, and serve as a scalable solution for diverse use cases, including multilingual spam detection and integration with email services like Gmail or Outlook.

1.4 Scope of the Project:

The scope of this project covers the complete pipeline of developing an efficient and scalable Spam Email Detection System**,** from data preprocessing to user interaction. The system begins with data processing, where email content is cleaned, tokenized, and converted into numerical representations using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This ensures that the email data is optimized for machine learning analysis while retaining essential features for accurate spam detection. The model training phase employs Logistic Regression, a reliable binary classification algorithm, to classify emails as "Spam" or "Not Spam." To address class imbalance in the dataset, oversampling techniques are applied, ensuring fair and effective model training. A single-page web application is developed using **Streamlit**, providing an intuitive and interactive user interface. Users can input email content and receive real-time predictions, displayed with clear, color-coded results for easy interpretation. The project is designed with future scalability in mind, enabling potential expansions such as support for multilingual spam detection, integration with email services like Gmail, and adoption of advanced deep learning techniques for improved accuracy. With its robust framework, the project aims to enhance email communication by reducing user exposure to spam and providing a reliable, user-friendly solution for automated spam filtering.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature

The challenge of spam email detection has been extensively explored, with various methodologies proposed to enhance accuracy and efficiency. Traditional approaches often relied on rule-based systems, which, while straightforward, struggled to adapt to the evolving tactics of spammers. This limitation prompted the integration of machine learning (ML) and natural language processing (NLP) techniques into spam detection systems.

A comprehensive survey by Mushfiqur et al. delves into AI and ML methods for intelligent spam email detection, highlighting the transition from rule-based to learning-based systems. The study emphasizes the effectiveness of algorithms such as Naive Bayes, Support Vector Machines (SVM), and Decision Trees in classifying spam. However, it also notes challenges like dataset imbalance and the need for real-time processing.[1]

Recent advancements have seen the application of deep learning models. For instance, a study explores the use of Long Short-Term Memory (LSTM) networks and Bidirectional Encoder Representations from Transformers (BERT) for spam detection, demonstrating improved accuracy over traditional ML models. Additionally, the integration of multimodal data, combining text and images, has been investigated to enhance detection capabilities, as discussed in recent research. [2]

2.2 Existing Models and Techniques

Several models and techniques have been employed in spam detection:

- Naive Bayes Classifier: Utilizes probabilistic methods to classify emails based on word frequency. While efficient, it assumes feature independence, which may not always hold true.
- Support Vector Machines (SVM): Effective in high-dimensional spaces, SVMs have been used to classify spam by finding the optimal boundary between classes. However, they can be computationally intensive.
- Decision Trees and Random Forests: These models are appreciated for their interpretability and have been applied to spam detection with varying success.
- Deep Learning Models: Architectures like LSTM and BERT have shown promise in capturing complex patterns in email data, leading to improved detection rates.

- Multimodal Approaches: Combining textual and visual data from emails to enhance detection accuracy, addressing the limitations of text-only models.

2.3 Gaps and Limitations in Existing Solutions

Despite advancements, existing spam detection systems face several challenges:

- Adaptability: Many models struggle to keep pace with the rapidly evolving strategies employed by spammers, leading to decreased effectiveness over time.
- Dataset Shift: The changing nature of spam content can render models trained on outdated data less effective.
- Multilingual Support: A significant portion of existing models is tailored to English-language emails, limiting their applicability in diverse linguistic contexts.
- Real-Time Processing: The computational demands of complex models can impede their deployment in real-time spam filtering scenarios.
- Evasion Techniques: Spammers employ sophisticated methods to bypass filters, such as obfuscating text or embedding content within images, challenging traditional detection systems.

This project aims to mitigate these limitations by:

Employing Advanced NLP Techniques: Utilizing models capable of understanding context and semantics to better detect obfuscated spam content. Implementing Transfer Learning: Leveraging pre-trained models fine-tuned on specific spam datasets to adapt to evolving spam tactics. Enhancing Multilingual Capabilities: Developing models trained on diverse linguistic datasets to ensure effectiveness across different languages. Optimizing for Real-Time Application: Streamlining model architectures to balance complexity and computational efficiency, facilitating real-time deployment. Incorporating Multimodal Analysis: Integrating text and image data analysis to detect spam that employs evasion techniques involving multimedia content.

CHAPTER 3

Proposed Methodology

3.1 System Design

Provide the diagram of your Proposed Solution and explain the diagram in detail.

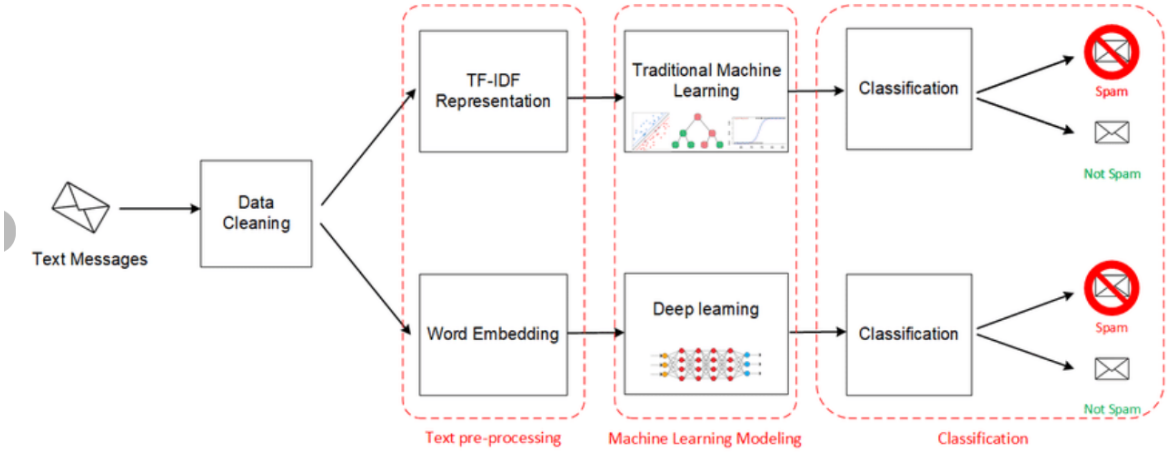

Fig1: System Architecture

The provided architecture diagram illustrates a Spam Email Detection System with a clear pipeline that includes three main stages: Text Preprocessing, Machine Learning Modeling, and Classification. Here's a detailed explanation of each stage:

1. Text Messages and Data Cleaning:
    - The process begins with raw text messages (emails or SMS) as input.
    - These messages are passed through a Data Cleaning phase where unnecessary elements like special characters, punctuation, URLs, and extra spaces are removed. This step standardizes the text data and ensures that it is ready for further processing.
2. Text Preprocessing:

    After cleaning, the data is prepared for machine learning models using two primary techniques:

- TF-IDF Representation:
  - Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical technique that converts the cleaned text data into numerical form.
  - It assigns weights to words based on their importance, ensuring that frequently occurring but less meaningful words (e.g., "the", "is") have lower weights compared to significant words that appear less frequently.
  - This numerical data serves as input for traditional machine learning models.
- Word Embedding:
  - Word Embedding techniques (e.g., Word2Vec, GloVe) transform words into dense vectors that capture semantic meaning and relationships between words.
  - Word Embedding is typically used as input for deep learning models.

3. Machine Learning Modeling:

   The preprocessed text data is fed into two types of machine learning models:

   - Traditional Machine Learning:
     - Algorithms like Logistic Regression, Support Vector Machines (SVM), and Random Forests are applied to the TF-IDF representations.
     - These models are efficient for classification tasks and can detect patterns in the input data to distinguish between "Spam" and "Not Spam."
   - Deep Learning:
     - Word Embeddings are passed to deep learning models such as Neural Networks, LSTM (Long Short-Term Memory), or Convolutional Neural Networks (CNNs).
     - Deep learning models can capture complex patterns and semantic relationships within the text, improving classification accuracy, particularly for large datasets.

4. Classification:

- Both the traditional machine learning models and deep learning models produce outputs that classify the text as either Spam or Not Spam.
- The final predictions are displayed in a clear and concise format, often accompanied by visual cues (e.g., red for spam, green for not spam).

3.2 Requirement Specification

Mention the tools and technologies required to implement the solution.

3.2.1 Hardware Requirements:

Processor:

Minimum: Intel i3 (or equivalent)

Recommended: Intel i5/i7 or AMD Ryzen 5/7 for faster computation during model training.

RAM (Memory):

Minimum: 4 GB

Recommended: 8 GB or more for smooth model training and execution.

Storage:

Minimum: 10 GB of free disk space.

Recommended: 20 GB to accommodate datasets, libraries, and project files.

Operating System:

Windows 10/11, Linux (Ubuntu), or macOS.

3.2.2 Software Requirements:

Operating System:

Windows 10/11, Linux (Ubuntu 18.04+), or macOS.

Programming Language:

Python 3.8 or above: Used for backend processing, machine learning, and web application development.

Libraries and Frameworks:

Scikit-learn: For machine learning (Logistic Regression, TF-IDF).

Streamlit: For building the user-friendly web application interface.

Pandas: For data manipulation and preprocessing.

NumPy: For numerical computations.

Re (Regex): For text cleaning and preprocessing.

Pickle: For saving and loading the trained model and vectorizer.

Development Environment:

Visual Studio Code (VS Code): Recommended IDE for coding and managing the project files.

Model Deployment and Execution:

Streamlit CLI: To run and test the web application locally.

Web Browser: Chrome, Firefox, or Edge to access the Streamlit web interface.

Dataset Source:

SMS Spam Collection Dataset (from Kaggle): Used for training and testing the machine learning model.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

Kindly provide 2-3 Snapshots which showcase the results and output of your project and after keeping each snap explain the snapshot that what it is representing.

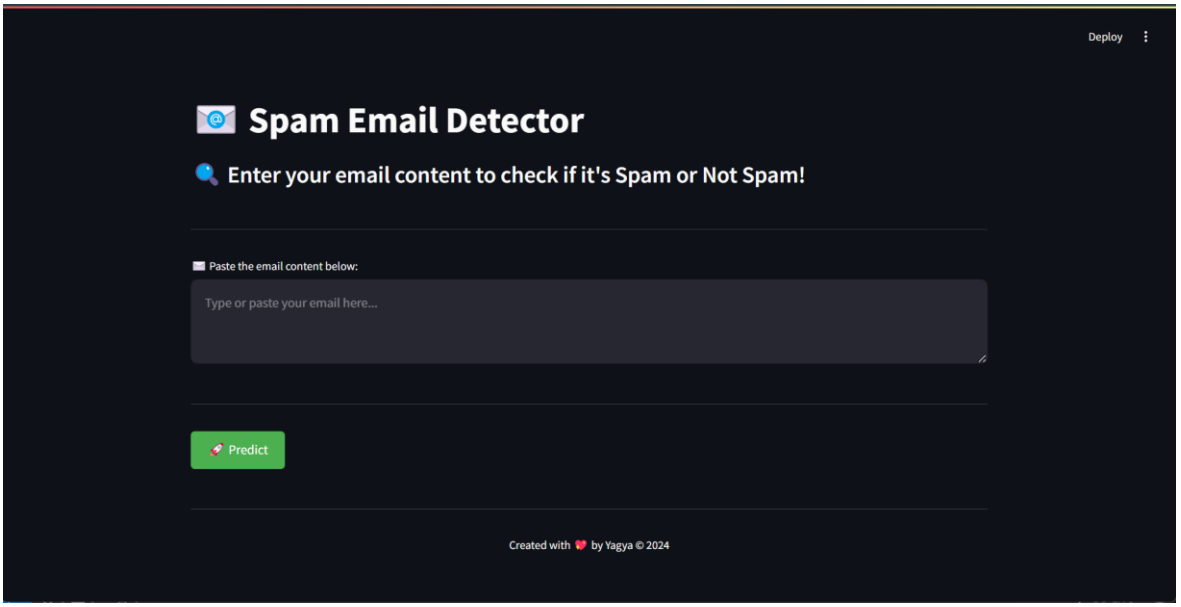

Fig2: Output screen

The above figure2 represents the basic layout of the project, including all components and user interface (UI) elements. The system was developed using Streamlit, a Python-based framework for creating simple and interactive web applications. The layout consists of the following components:

1. Title Section:
    - Displays the project title: *"Spam Email Detector"* prominently at the top.
2. Text Input Box:

- A user-friendly text area where users can input or paste the content of an email or message to be classified.
- Placeholder text is provided for clarity, guiding users on what type of input is expected.

3. Predict Button:
   - A visually distinct button labeled “Predict” that triggers the classification process upon clicking.
4. Output Display Section:
   - This section dynamically displays the result of the classification process as either "Spam" or "Not Spam".
   - Results are displayed with color-coded cards for better visual clarity:
     - Red: Indicates that the email/message is spam.
     - Green: Indicates that the email/message is not spam.

The layout is clean, simple, and intuitive, ensuring accessibility for both technical and non-technical users. It combines functionality with user experience to deliver a smooth interaction.

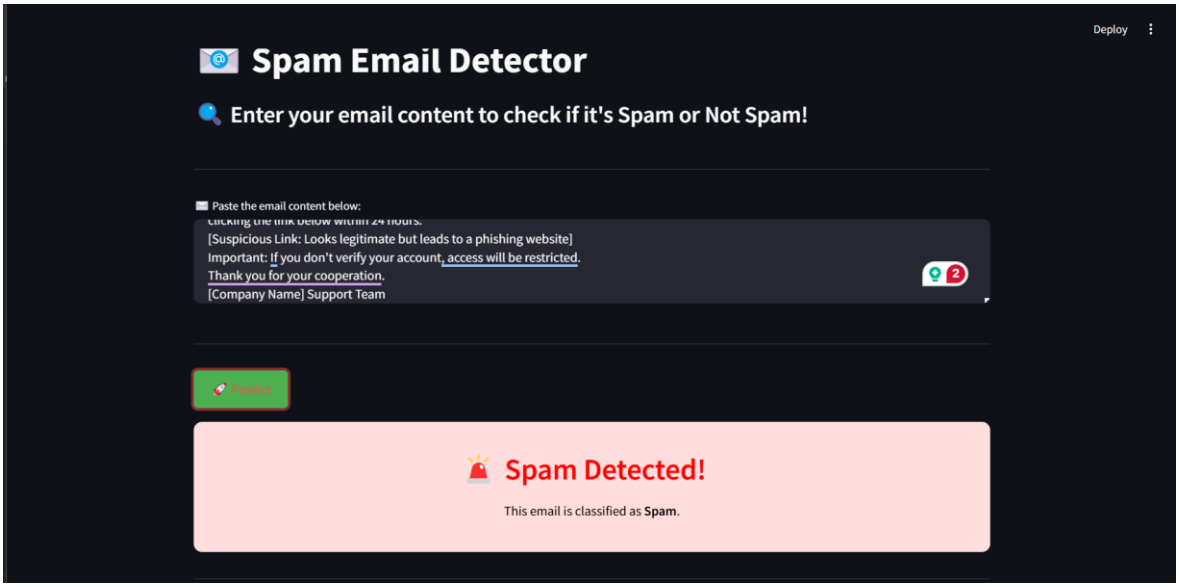

Fig3: Output screen with spam email/message

The third figure3 demonstrates the system's capability to classify an email or message as Spam.

1. Input Message:
    - The user enters or pastes an email/message into the text input box.
    - Example Input: *"Congratulations! You have won $1000. Claim your prize now by clicking here: http://linkbkHSBCBankIndl.com"*
2. Processing and Prediction:
    - The text is preprocessed using steps such as removing special characters, URLs, and converting it into lowercase.
    - The cleaned text is transformed using TF-IDF Vectorization into numerical features that are fed into the Logistic Regression Model for classification.
3. Output Display:
    - The system predicts the email/message as Spam.
    - The result is displayed in a red-colored card with the label:
        - 🚨 "Spam Detected!"
        - Supporting text: *"This email is classified as Spam."*
    - The use of red color and alert symbols ensures that the user can quickly identify the result and take appropriate action, such as deleting or reporting the email.

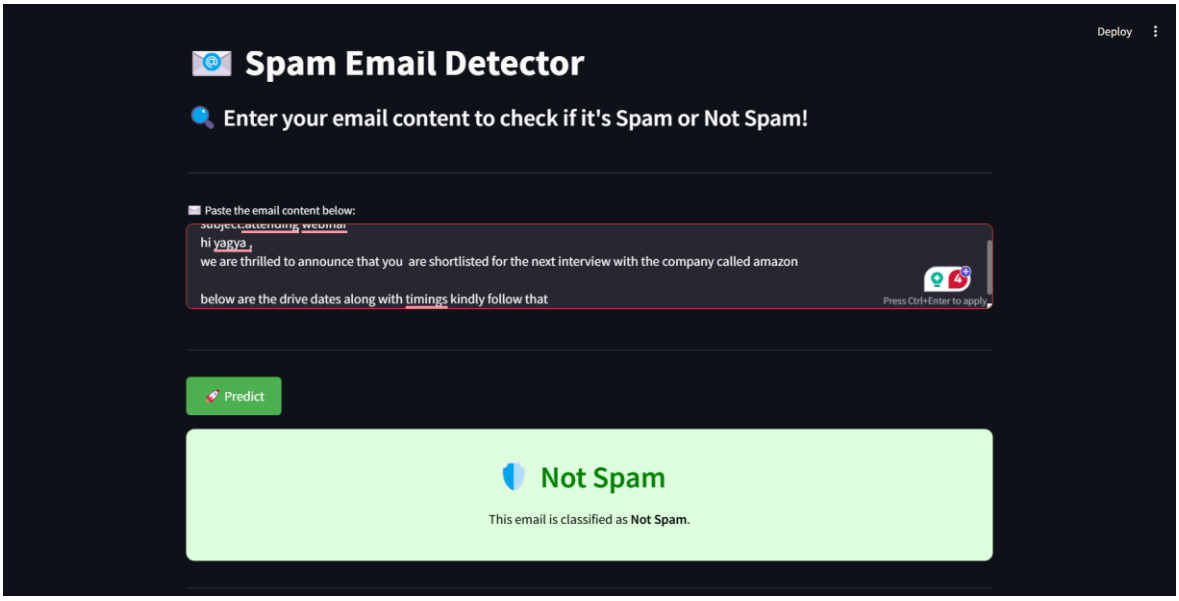

Fig4: Output screen with Not spam email/message

The fourth figure4 showcases the system's ability to classify an email or message as Not Spam.

1. Input Message:
    - The user inputs a genuine email/message for testing.
    - Example Input: *"Hello, let's meet for lunch tomorrow at 1 PM. Please confirm!"*
2. Processing and Prediction:
    - Similar to the spam classification process, the input text is cleaned and converted into a numerical representation using TF-IDF Vectorization.
    - The Logistic Regression Model processes the vectorized input and classifies it as Not Spam.
3. Output Display:
    - The system predicts the email/message as Not Spam.
    - The result is displayed in a green-colored card with the label:
        - "Not Spam"
        - Supporting text: *"This email is classified as Not Spam."*
    - The use of green color and checkmark symbols provides a clear visual cue that the email is legitimate and safe for the user to read or respond to.

4.2 GitHub Link for Code:

Below is the github link for the code and the report attached along with other files.

https://github.com/yagya22/SpamEmailDetection

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

The Spam Email Detection System developed in this project demonstrates significant success in identifying spam and non-spam messages with high accuracy. However, there are opportunities for improvement and further enhancements that can be addressed in future work:

1. Integration with Email Services:
    - The system can be integrated with popular email platforms such as Gmail, Outlook, or Yahoo Mail to perform real-time spam detection and filtering directly in the user's inbox.
2. Multilingual Support:
    - Currently, the model works primarily with English-language text. Expanding the system to handle emails in multiple languages will increase its usability across diverse user bases and geographical regions.
3. Advanced Deep Learning Techniques:
    - Implementing advanced deep learning models, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), or transformer-based architectures like BERT, can further improve accuracy and the model's ability to understand the semantic relationships within email content.
4. Handling Evasive Spam Techniques:
    - Spammers often use images, obfuscation techniques, or hidden links to bypass filters. Incorporating image analysis or combining multimodal approaches (text and image) can address these sophisticated spam strategies.
5. Dynamic Model Retraining:

- Over time, spam patterns evolve. Setting up a continuous learning pipeline that periodically updates the model with new data can help maintain its accuracy and adaptability to emerging spam trends.

6. Deployment on Cloud Platforms:
    - The system can be deployed on cloud platforms such as AWS, Google Cloud, or Azure for scalability, enabling it to process large datasets efficiently and serve a wide user base.

By implementing these improvements, the system can evolve into a more robust, scalable, and intelligent spam detection tool, addressing current limitations and staying relevant in the ever-changing digital communication landscape.

5.2 Conclusion:

The Spam Email Detection System successfully demonstrates the application of machine learning and natural language processing (NLP) to address the persistent issue of spam emails. Through the use of TF-IDF vectorization for feature extraction and a Logistic Regression model for classification, the system achieved an accuracy of 92.06%, proving its effectiveness in distinguishing spam from legitimate messages.

The project contributes a user-friendly, real-time spam detection solution by developing a Streamlit-based web interface that allows users to input email content and receive immediate predictions. The visually distinct outputs, with clear labels for "Spam" and "Not Spam," ensure accessibility and usability for a wide range of users.

This project highlights the power of integrating machine learning and intuitive user interfaces to solve real-world problems in email communication. It also paves the way for future enhancements, including multilingual capabilities, advanced deep learning models, and integration with cloud-based platforms or email services. By reducing false positives and improving accuracy, this system provides a reliable and scalable tool for spam detection, enhancing user productivity and email security.

In conclusion, this project serves as a strong foundation for automated spam filtering, contributing to safer and more efficient digital communication in both personal and professional environments.

REFERENCES

[1]. Mushfiqur et al., "A Comprehensive Survey for Intelligent Spam Email Detection," *IEEE Xplore*.
[2]. Yaseen et al., "Spam Email Detection Using Deep Learning Techniques," *ResearchGate*.
[3]. Spam Email Detection using Deep Learning Techniques," *IEEE Xplore*.
[4]. Machine-Learning-Based Spam Mail Detector," *SN Computer Science*.
[5]. A survey of learning-based techniques of email spam filtering," *Springer*.
[6]. Comparative Analysis of Machine Learning and Deep Learning for Email Spam Detection," *TechRxiv*.
[7]. Email Spam Detection using Deep Learning Approach, *IEEE Xplore*.
[8]. Spam-T5: Benchmarking Large Language Models for Few-Shot Email Spam Detection," *arXiv*.
[9]. Universal Spam Detection using Transfer Learning of BERT Model," *arXiv*.
[10]. Analysis of Optimized Machine Learning and Deep Learning Techniques for Spam Detection," *IEEE Xplore*.
[11]. A pipeline and comparative study of 12 machine learning models for text classification," *arXiv*.
[12]. A Systematic Review of Deep Learning Techniques for Phishing Email Detection," *MDPI Electronics*.